AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1-16. (Cancelled)

17. (Currently Amended) A method for rendering three-dimensional scenes on a display of a computing device, the method comprising:

rendering the three dimensional scenes, by the computing device, ~~by means of~~via ray-tracing;

establishing, by the computing device, a ray-tracing acceleration structure having a plurality of objects;

traversing, by the computing device, a ray through the acceleration structure until an object of intersection with the ray is identified;

performing a first intersection computation between the object of intersection and the ray;

storing, by the computing device, the object of intersection in a list of objects that have been intersected by the ray; and

after the object of intersection is stored in the list:

searching the list, by the computer device, for the object of intersection~~.~~, and

preventing, by the computing device, further performing of intersection computations between the object of intersection and the ray after performing the first intersection computation.

18. (Previously Presented) The method of claim 17 further comprising:

determining, by the computing device, whether an affirmative decision is possible as to whether a triangle within the ray-tracing acceleration structure overlaps a node of the acceleration structure, the decision being based at least in part on comparisons of the vertices of the triangle with the vertices of the node; and

when the affirmative decision is not possible, deciding, by the computing device, by making a conservative decision as to whether the triangle overlaps the node.

19. (Previously Presented) The method of claim 17 further comprising:

computing, by the computing device, a three dimensional ray space occupied by the ray according to the ray's increasing distance from a virtual source;

traversing, by the computing device, the ray space; and

processing, by the computing device, selected objects identified within the ray space.

20. (Previously presented) The method of claim 19, wherein the ray space is a selected one of a cone or a pyramid, and the method further comprises processing, by the computing device, objects within the ray space, and ordering the objects according to the objects' virtual distances from the virtual source of the ray.

21. (Currently Amended) The method of claim 19, wherein the ray-tracing acceleration structure further comprises at least one acceleration structure node, with each node being associated with a region of node space, and the method further comprises:

storing, by the computing device, simplified geometry data associated with ~~each~~ respective acceleration structure nodes, the respective simplified geometry data ~~which omits~~ omitting one or more triangles from geometry data for the node; and

computing, by the computing device, a ray-object intersection, using the simplified geometry data to compute the ray-object intersection when the ray space overlaps a majority of the node space.

22. (Previously Presented) The method of claim 17 further comprising:

subdividing, by the computing device, the plurality of objects into an n-level hierarchy of objects and sub-objects;

classifying, by the computing device, the objects as primitive or non-primitive;

traversing the ray through the acceleration structure until an intersected object is identified; and

after traversing the ray,

if the identified object is not a primitive object, storing, by the computing device, affine transformation data which transforms the object and ray into a coordinate system locally associated with the identified object, and traversing the local coordinate system and storing affine transformation data, recursively, until the ray intersects with a primitive object; and

if the identified object is a primitive object, storing, by the computing device, transformation data which transforms the primitive object and ray into a normalized object space locally associated with the identified object, intersecting the transformed ray with the transformed identified object, and storing, by the computing device, ray-object intersection data.

23. (Previously presented) The method of claim 17 further comprising:

subdividing, by the computing device, the plurality of objects into an n-level hierarchy of objects and sub-objects;

recursively instantiating, by the computing device, a plurality of sub-objects, the sub-objects having a similar geometry; and

building, by the computing device, a next level object by, at least in part, using the plurality of instantiated sub-objects with similar geometry.

24. (Previously Presented) An apparatus for rendering three-dimensional scenes, the apparatus comprising:

a processor;

a tangible computer-accessible storage medium operatively coupled to the processor;

a ray-tracing acceleration structure stored in the storage medium, the acceleration structure having object data for a plurality of objects; and

a ray-casting module operatively configured to traverse rays through the acceleration structure and return ray-object intersection data, the ray-casting module comprising a decision unit operatively configured such that, when ray-object intersection data has been computed for a given ray and a given object, the decision unit records that the given ray has intersected the given object and prevents additional ray-object intersection computations from being carried out for the given ray and the given object.

25. (Previously presented) The apparatus of claim 24 wherein the ray-casting module further comprises an acceleration structure computation unit operatively configured to:

determine if an affirmative decision is possible as to whether a triangle within the ray-tracing acceleration structure overlaps a node of the acceleration structure, the decision being based at least in part on comparisons of the vertices of the triangle with the vertices of the node; and

if the affirmative decision is possible, store the affirmative decision as part of establishing the acceleration structure.

26. (Previously presented) The apparatus of claim 25 wherein, the acceleration structure computation unit is further configured to compute, if the affirmative decision is not possible, a conservative decision as to whether the triangle overlaps the node and store the conservative decision as part of establishing the acceleration structure.

27. (Previously Presented) The apparatus of claim 24 wherein the ray-casting module further comprises a traversal unit, the traversal unit operatively configured to:

compute a three dimensional ray space occupied by the ray according to the ray's increasing distance from a virtual source; and

traverse the ray space through the acceleration structure from the virtual source of the ray through an increasing virtual distance from the virtual source of the ray.

28. (Previously presented) The apparatus of claim 27 wherein the ray space traversed is a selected one of a cone or a pyramid.

29. (Previously presented) The apparatus of claim 27 wherein the acceleration structure further comprises at least one acceleration structure node, each node being associated with a region of node space; and wherein the ray casting module is further configured to store simplified geometry data associated with each acceleration structure node which omits one or more triangles from geometry data for the node; and, when the ray space overlaps a majority of the node space, compute ray-object intersection using the simplified geometry data.

30. (Previously Presented) The apparatus of claim 24 wherein the ray-casting module further comprises:

a matrix unit;

a transformation unit; and

a ray casting unit,

wherein the matrix unit is operatively configured to send affine transformation matrices of object data to the transformation unit, each affine transformation matrix being a selected one of an object-space transformation matrix and a normalized space transformation matrix, and the transformation unit is further operatively configure to receive the ray from the ray casting unit and transform the ray into the object's affine transformation matrix's coordinate system.

31. (Previously Presented) The apparatus of claim 30 wherein the ray-casting module further comprises:

an acceleration structure computation unit operatively configured to subdivide the plurality of objects into an n-level hierarchy of objects and sub-objects and store data classifying the objects as primitive or non-primitive;

a traversal unit operatively configured to traverse the ray through the acceleration structure until a potential ray-object intersection is identified; and

an intersection computation unit;

wherein, the transformation unit is operatively configured to send the object's affine transformation matrix data and the transformed ray data to the transversal unit for recursive processing until the ray intersects with a primitive object, if the object identified by the traversal unit is not a primitive object; and

wherein, the transformation unit is operatively configured to send the object's affine transformation matrix data and the transformed ray data to the intersection computation unit, if the object identified by the traversal unit is a primitive object.

32. (Currently Amended) An article of manufacture comprising a tangible, non-transitory computer-accessible storage medium for digitally storing a plurality of programming instructions designed to program an apparatus to render three-dimensional scenes ~~by means of~~via ray-tracing, wherein the programming instructions, if executed by the apparatus, enable the apparatus to:

establish ray tracing acceleration structures having a plurality of objects; and

for a given ray, and for an object for which an intersection computation has been performed with the given ray:

record that the intersection computation has been performed for the given ray and the object; and

prevent further performing of intersection computations between the object and the given ray.

33. (Previously presented) The article of manufacture of claim 32 wherein the programming instructions, if executed, further enable the apparatus to:

determine if an affirmative decision is possible as to whether a triangle within the ray-tracing acceleration structure overlaps a node of the acceleration structure, the decision being based at least in part on comparisons of the vertices of the triangle with the vertices of the node; and

if the affirmative decision is not possible, decide by making a conservative decision as to whether the triangle overlaps the node.

34. (Previously Presented) The article of manufacture of claim 32 wherein the programming instructions, if executed, further enable the apparatus to:

compute a three dimensional ray space occupied by the given ray according to the ray's increasing distance from a virtual source;

traverse the ray space through the acceleration structure; and

process selected objects identified within the ray space.

35. (Previously Presented) The article of manufacture of claim 34 wherein the ray space traversed by the ray is a selected one of a cone or a pyramid, and the programming instructions, if executed, further enable the apparatus to process the selected objects within the ray space, and order the selected objects according to the selected objects' virtual distances from the virtual source of the ray.

36. (Previously presented) The article of manufacture of claim 34 wherein the acceleration structures have at least one acceleration structure node, each node being associated with a region of node space; and wherein the programming instructions, if executed, further enable the apparatus to:

store simplified geometry data associated with each acceleration structure node which omits one or more triangles from geometry data for the node; and

when the ray space overlaps a majority of the node space, compute ray-object intersection, using the simplified geometry data.

37. (Previously Presented) The article of manufacture of claim 32 wherein the programming instructions, if executed, further enable the apparatus to:

subdivide the plurality of objects into an n-level hierarchy of objects and sub-objects;

classify the objects as primitive or non-primitive;

traverse the given ray through the acceleration structure until an intersected object is identified,

if the identified object is not a primitive object, store affine transformation data which transforms the identified object and ray into a coordinate system locally associated with the identified object, and traverse the transformed ray through the local coordinate system and store affine transformation data, recursively, until the ray intersects with a primitive object; and

if the identified object is a primitive object, store transformation data which transforms the primitive object and ray into a normalized object space locally associated with the identified object, intersect the transformed ray with the transformed identified object, and store ray-object intersection data.

38. (Previously presented) The article of manufacture of claim 32 wherein the programming instructions, if executed, further enable the apparatus to:

subdivide the plurality of objects into an n-level hierarchy of objects and sub-objects;

recursively instantiate a plurality of sub-objects, the sub-objects having a similar geometry; and

build a next level object by, at least in part, using the plurality of instantiated sub-objects with similar geometry.

39. (Previously Presented) A ray casting processor for rendering dynamic three-dimensional scenes, the processor comprising:

a traversal unit, the traversal unit being coupled to a plurality of node caches for storing acceleration structure node data, the node data structured to improve accessibility according to ray coherence, the node caches arranged in an n-level hierarchy;

a list unit, the list unit being coupled to a plurality of list caches for storing lists of object addresses for objects that have already been intersected by a ray, the list caches arranged in an n-level hierarchy, the list unit having a decision unit, the decision unit configured to prevent objects whose object addresses are stored in the list from being intersected again by the ray; and

a matrix loading unit, the matrix loading unit being coupled to a plurality of matrix caches for storing affine transformations in the form of matrices, the matrix caches arranged in an n-level hierarchy.